Möbius Cellular Automata Scarves

Elisabetta A. Matsumoto[1], Henry Segerman[2] and Fabienne Serriere[3]

[1]School of Physics, Georgia Institute of Technology, Atlanta, USA; sabetta@gatech.edu
[2]Department of Mathematics, Oklahoma State University, Stillwater, USA; henry@segerman.org
[3]KnitYak, Seattle, USA; fabienne@fabienne.us

Abstract

In 2015, the third author launched a successful Kickstarter campaign to fund the purchase of an industrial knitting machine. The Kickstarter rewards were scarves, each procedurally knitted in a unique two-colour pattern: the output of a elementary cellular automaton. The scarves are double knit (double bed jacquard in machine knitting parlance), meaning that the scarf has two layers of different colours, that swap positions back and forth to produce the pattern on the scarf. This implies that the pattern on the back side of the scarf is the colour reverse of the pattern on the front side. A corresponding "inverse" elementary cellular automaton produces the pattern on the back. We wondered if it would be possible to produce a Möbius strip scarf, in which the front becomes the back whilst seamlessly continuing the development of a single elementary cellular automaton. This paper describes our discoveries.

Elementary Cellular Automata

An elementary cellular automaton [1] is a one-dimensional cellular automaton with two states, for which the state c_i^{n+1} of a cell in position i at generation $n + 1$ is determined by c_{i-1}^n, c_i^n and c_{i+1}^n in the previous generation n. There are eight possible states of the three consecutive cells that determine the state of the middle cell in the next generation, so there are $2^8 = 256$ possible elementary cellular automata ("rules"). These are numbered by reading the eight output bits as a binary number. See Figure 1. The rules act on binary strings, which in this paper are assumed to be cyclic, avoiding special cases at the ends of a string.
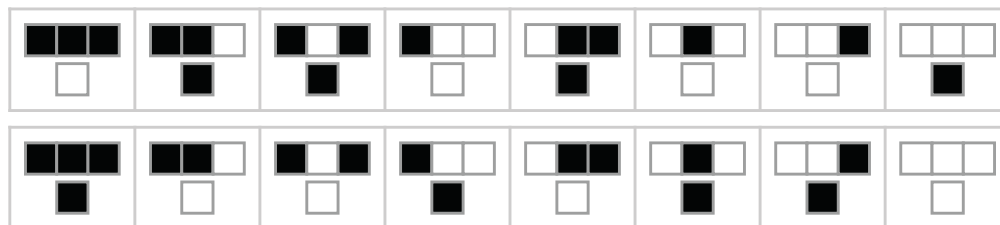

Figure 1: *The eight possible states of three consecutive cells determine the state of the middle cell in the next generation. Two rules are shown here. The first is numbered as* 01101001 *in base 2, which is* 105 *in base 10. The second is numbered as* 10010110 *in base 2, which is* 150 *in base 10.*

The Scarf Inverse of a Cellular Automaton

Figure 2: *Front and back of double knit.*

In double knitting, two colours of yarn are used, each colour making a planar knitted surface, one on the front of the piece (eg. black) and one on the back (eg. white). Colour work in double knitting exchanges stitches between the back surface and the front surface, swapping the two colours. Therefore, if we see a binary string in a row of double knitting, the opposite side of the work will show that binary string both reversed and with colours interchanged. See Figure 2. This motivates the following definitions:

The *mirror* of a binary string is the result of reversing the string. For example, 01101 becomes 10110. The *complement* of a binary string is the result of exchanging all 0s and 1s, which we denote by a bar. For example, $\overline{01101} = 10010$. The combination of these two operations produces the mirrored complement of a string, which we call the *scarf inverse*, and denote with a tilde. So, for example, $\widetilde{01101} = 01001$. If we view a row of stitches of a double knit scarf as a binary string, then applying the scarf inverse results in the binary string for the other side of the same row of stitches.

Given a scarf whose subsequent rows are generated by an elementary cellular automaton, A, one can ask if there is a cellular automaton, A^* say, that generates the subsequent rows of the other side of the scarf. That is, is there a rule A^* so that for every binary string s, $\widetilde{A(s)} = A^*(\widetilde{s})$? The answer is yes: A is a map from the eight length three input strings to $\{0, 1\}$. For each such length three input string s, we define $A^*(s) = \overline{A(\overline{s})}$. This produces the desired behaviour. We call A^* the *scarf inverse* of A.

We are interested in rules with the property that $A = A^*$, since then the same rule simultaneously runs on "both sides" of a Möbius strip scarf. There are 16 such rules, whose numbers can be generated as the binary strings $abc\overline{b}d\overline{c}\overline{d}a$, for $a, b, c, d \in \{0, 1\}$. In decimal, the possibilities are 23, 29, 51, 57, 71, 77, 99, 105, 150, 156, 178, 184, 198, 204, 226, and 232. See Figure 3 for sample runs of these rules, all starting from the same randomly chosen seed. Most of the rules produce

Figure 3: *16 scarf inverse invariant rules with the same random seed.*

uninteresting patterns. Thankfully however, we find two rules, 105 and 150, shown in Figure 1, that are both invariant under the scarf inverse operation, and give interesting patterns. We focus on these rules.
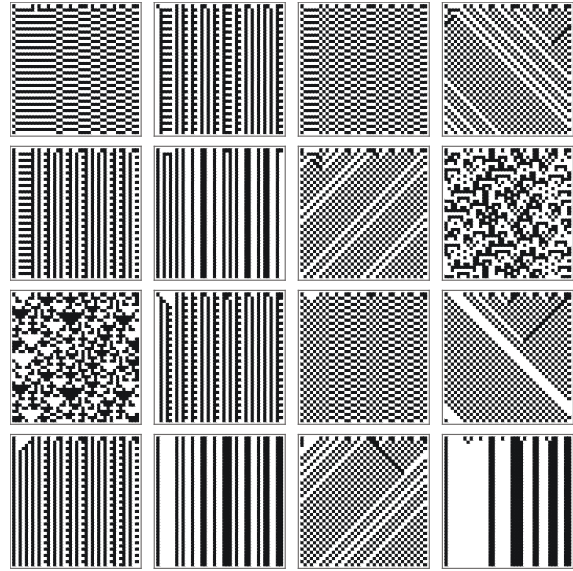
Finding a Cycle

Having determined the rules we want to use, we next need to actually produce a pattern. We must produce a binary string corresponding to a row of stitches in the scarf, that when acted on by our rule, produces a sequence of strings that eventually returns back to the scarf inverse of the initial binary string (so that we can stitch the end of the scarf to the start and produce a Möbius strip). Moreover, in order to produce a reasonable scarf, we need each row to be around 100 stitches wide and the cycle length to be around 1000 rows. Initially, this seems to be an impossible problem. Given a random 100 digit binary string, it would seem to be extremely improbable that after applying a rule around 1000 times, we would arrive at its scarf inverse. We are guaranteed to eventually fall into a loop, since there are a finite number of possible binary strings, but there is no guarantee that such a loop would contain any pairs of scarf inverse strings. Moreover, the search space of possible initial binary strings is computationally intractable, so we cannot directly brute force the problem. Our solution is to require that our initial binary string is itself invariant under the scarf inverse operation. This implies that under our rule, each subsequent string is also invariant under the scarf inverse operation. Now we need only find a loop. It is perhaps a little disappointing that such a loop could be stitched as either a Möbius strip or an annulus, both respecting the cellular automaton. This also means that there is a symmetry in each row that is not essential to the concept, although invariance of each row under the scarf inverse is not immediately visually obvious. In any case, our method has the immense advantage that it works!
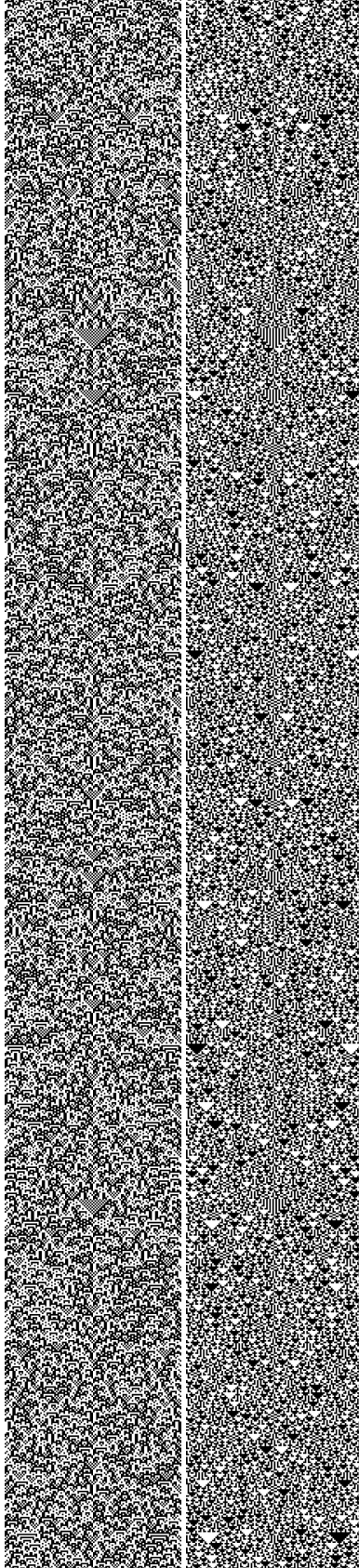
Figure 4: *Scarf patterns.*

Our algorithm to try to find a loop is as follows. We take as input a *scarf width*, which is another name for the length of the binary strings we will use. We require that this is an even number, since all of our strings are invariant under the scarf inverse operation. We start with an initial random binary string, formed by taking a random string of half the scarf width, and appending a mirrored complementary copy. That is, given any binary string s, we form the string $s\widetilde{s}$, which is scarf inverse invariant. Next, we generate a sequence of some large number of iterations of our cellular automaton rule, applied to $s\widetilde{s}$. To avoid any initial tail, we throw away an initial segment of the list, of length randomly chosen between 1,000 and 2,000. We then take the next string in the sequence, and search the remainder of the list for a repeat of this string.

Results

We ran the above algorithm for all different even scarf widths from 70 through 130. As far as we can tell, in the vast majority of cases, if we are able to find a loop then we find the *exact same* loop every time we run the algorithm. That is, for randomly chosen initial binary strings, we seem to fall into the same loop for a given scarf width. Moreover, the loops for rules 105 and 150 seem to have the same cycle lengths. These cycle lengths are listed in the table below. For each scarf width, if we could not find a loop in 33,600,000 iterations (slightly larger than 2^{25}), this is indicated by a dash in the table below. These cases must eventually loop, but we don't know when.

scarf width	70	72	74	76	78	80	82	84
cycle length	$2^{13} - 2$	56	58,254	$2^{11} - 4$	$2^{13} - 2$	48	$2^{11} - 2$	$2^8 - 4$

scarf width	86	88	90	92	94	96	98
cycle length	$2^8 - 2$	$2^8 - 8$	$2^{13} - 2$	$2^{13} - 4$	$2^{24} - 2$	16	$2^{22} - 2$

scarf width	100	102	104	106	108	110	112
cycle length	$2^{12} - 4$	$2^9 - 2$	168	-	$2^{11} - 4$	$2^{21} - 24$	112

scarf width	114	116	118	120	122	124	126	128	130
cycle length	$2^{10} - 2$	$2^{16} - 4$	-	120	-	124	126	64	126

When seeded with a highly symmetric initial binary string, such as half zeroes and half ones, we can fall into different loops from the ones we get from randomly chosen seeds, and so the cycle length changes for several of the seed lengths. For initial binary strings $00\ldots0011\ldots11$ and $11\ldots1100\ldots00$, the changes to the cycle length occur in two ways: Either the cycle length compared with the random seed is halved for both rules – this is the case for widths 72, 76, 80, 84, 88, 92, 100, 104, 108, 112, 120, and 124; or the cycle length is only halved for rule 150 – this is the case for widths 70, 78, 82, 86, 90, 102, 114, 126, and 130. For an initial binary string $0101\ldots0101$, we immediately get a checkerboard pattern for rule 105 and alternating vertical stripes for rule 150, so for this initial string the cycle lengths are all 2 for rule 105 and 1 for rule 150.

Figure 5: *Scarf coming off the machine.*

We can explain why rules 105 and 150 generally have the same cycle lengths: Let us write rule 105 as A and rule 150 as B. By inspecting Figure 1, we can see that for any input string s, $B(s) = \overline{A(s)} = A(\overline{s})$, so $A(s) = B(\overline{s})$. Then $A(A(s)) = B\left(\overline{A(s)}\right) = B(B(s))$. So the sequences of binary strings generated by A and B starting from the same initial string are complementary at odd iterations and identical at even iterations. The cycle lengths can therefore only differ if one happens to hit a cycle at an odd iteration, while the other has reached the complement of its initial string. In this unlikely event, the cycle lengths differ by a factor of two.

There are however a number of other questions that arise from looking at this data that we do not have answers to. Why does there seem to be only one long loop for each scarf width? Why do many of the cycle lengths seem to be just under a power of two? We also noticed that some of the cycle lengths are the same as the scarf width. Again we have no explanation.

These questions aside, we obtained some good candidate designs, with cycle lengths near 1000. We chose the two designs with scarf width 114, cycle length $2^{10} - 2 = 1022$. These are shown in Figure 4 (rule 105 on the left and 150 on the right).

Knitted scarves typically have an aspect ratio of approximately 1/6, with the width about a third of a meter and the length around two meters. The gauge of the Stoll CMS 530 HP 7.2 multigauge industrial flat knitting machine used for our double knitted scarves after being steamed is 51 stitches per 10 cm wide (columns), 54 stitches per 10 cm tall (rows), with tension on the Stoll knitting machine set to 13.5 using 2/18 USA yarn count gauge merino yarn (approximately equivalent to 2/24nm gauge). Note that knitted stitches aren't square. Scarves produced by this knitting machine come out as planar fabric (Figure 5). In order to make Möbius (or annular) scarves one needs to graft the top and bottom of the scarf together, using a technique called Kitchenering, which uses a needle and yarn to create the topology of the knitted stitch between two different pieces of fabric. The left and center images in Figure 6 are renders of what the finished garments will look like after they have been grafted together. A of the photo finished rule 150 scarf is shown to the right.
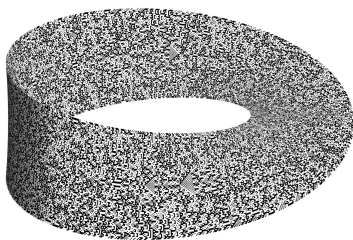
(a) *Render of Möbius scarf rule 105.*
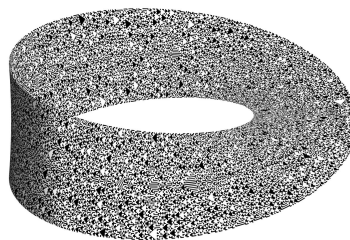
(b) *Render of Möbius scarf rule 150.*

(c) *Knitted Möbius scarf rule 150.*

Figure 6: *Möbius scarves.*

References

[1] Stephen Wolfram. *A New Kind of Science*. Wolfram Media Inc., Champaign, IL, USA, 2002.